Journal of Artificial Intelligence and Data Mining (JAIDM), Vol. 12, No. 1, 2024, 137-147.

Shahrood University of Technology

Journal of Artificial Intelligence and Data Mining (JAIDM)
Journal homepage: http://jad.shahroodut.ac.ir

Research paper

A Multi-View Model for Knowledge Graph Embedding in Link Prediction using GRU-RNN as Constraint Satisfaction Problem

Afrooz Moradbeigy and Farzin Yaghmaee*

Electrical and Computer Engineering, Semnan University,Semnan, Iran.

Article Info

Article History:
Received 25 November 2023
Revised 27 February 2024
Accepted 08 April 2024

DOI:10.22044/jadm.2024.13827.2498

Keywords:
Multi-view, Knowledge Graph Embedding, Constraint Satisfaction Problem, GRU-RNN.

**Corresponding author: f_Yaghmaee@semnan.ac.ir (F. Yaghmaee).*

Abstract

Knowledge graphs are widely used tools in the field of reasoning, where reasoning is facilitated through link prediction within the knowledge graph. However, traditional methods have limitations, such as high complexity or an inability to effectively capture the structural features of the graph. The main challenge lies in simultaneously handling both the structural and similarity features of the graph. In this study, we employ a constraint satisfaction approach, where each proposed link must satisfy both structural and similarity constraints. For this purpose, each constraint is considered from a specific perspective, referred to as a view. Each view computes a probability score using a GRU-RNN, which satisfies its own predefined constraint. In the first constraint, the proposed node must have a probability of over 0.5 with frontier nodes. The second constraint computes the Bayesian graph, and the proposed node must have a link in the Bayesian graph. The last constraint requires that a proposed node must fall within an acceptable fault. This allows for N-N relationships to be accurately determined, while also addressing the limitations of embedding. The results of the experiments showed that the proposed method improved performance on two standard datasets.

1. Introduction

The concept of a knowledge graph for improving web search results was introduced by Google in 2012. This graph structure has since been applied in question answering and recommendation systems [1,2,3]. The knowledge graph represents entities as nodes (head and tail) and relationships as edges (relation) in a directed manner, denoted by triplet (head (h), relation (r), tail (t)). Despite being incomplete, the knowledge graph has received significant research attention due to its potential to clarify cause-and-effect relationships [5,6,7,8]. Knowledge graph embedding (KGE) is a method used to extract knowledge from a graph. Embedding refers to the process of representing entities and relationships as low-dimensional vectors in a continuous vector space and embedding is used in various task [9,10]. New embedding methods are introduce for graph embedding to reduce the number of training data [11]. Knowledge graph embedding methods aim to encode the structured information contained in a knowledge graph into these vector representations [4].

Link prediction in knowledge graph involves identifying missing edges in the embedded graph [12]. There have been numerous methods proposed for link prediction, which can be grouped into three categories: translation-based, bilinear, and neural network-based methods [13]. Translation-based methods treat links as similarity or distance measures in the embedded space and include TransE and TransR [14,15]. Bilinear methods use mapping functions to calculate the quality of representations for triplets and include RESCAL, HOlE, and ComplEx [16,17,18]. Neural network methods, such as ConvE and ConvKB use deep learning networks to learn the relationships in the data [19,20]. However, many of these methods do

not take into account the structural relationships between nodes or the different types of entities and relations within the knowledge graph. Additionally, deep learning methods often result in gradient loss and inconsistent evaluation criteria, and do not account for the directness of the graph [21,22,23]. Gradient loss refers to the decrease in the gradient of a loss function, indicating a slow rate of change in the loss function. This phenomenon occurs when the rate of change becomes negligible. In the context of deep learning methods, gradient loss can pose challenges, as it leads to difficulties in optimizing the model parameters effectively.

The proposed method in this paper adopts deep learning algorithms due to their potential to provide effective solutions. To address the limitations of previous methods, the score function learned by the deep network is designed to consider the directness of the graph, the relationships between entities, and the graph structure. To accomplish this, the method employs multi-view learning to take into account various features of the graph simultaneously. To mitigate gradient loss, a GRU layer is added to an RNN. The mapping function considers multiple features and it model both 1-N and N-N relationships. This paper is organized as follows: Section 2 provides a review of relevant research literature, Section 3 describes the proposed method, Section 4 presents the experimental results, and the conclusion of the research is presented in Section 5.

2. Literature Review

The knowledge graph embedding models aim to learn the entities and relationships in a graph structure. The embedding models that utilize structured graph features, such as one of the early methods in the field [13], provide a vector representation for each entity and a mapping matrix for the relationships. However, this model has limitations as it does not allow for direct interaction between entities [17]. RESCAL presents a vector for each entity to capture its intrinsic meanings and uses an incidence matrix between pairs of entities to demonstrate their relationships [17]. Despite its capability to capture rich information, RESCAL requires a large number of parameters. The large number of parameters leads to complexity and significantly increases computation time. DistMult is a simplified version of RESCAL, but it sacrifices its representation power by limiting the relationship mapping to a diagonal matrix, reducing the number of parameters. Holographic embedding (HolE) combines the simplicity of DistMult and the expressiveness of RESCAL by representing both entities and relationships as vectors, but it fails to differentiate between the entities and relation [16].

The model TransE was introduced in 2013 as an adaptation of Word2Vec [14]. It operates based on a relationship (r) serving as the translation between two entities, h and t, in a triplet (h, r, t). TransE means that the vector numeric mapping (embedding) of entity t is placed near the embedding of entity h through relation r. The score function is defined as the distance between h and t. Although TransE demonstrated relatively better performance compared to prior models, it had limitations in handling many-to-many (N-N) relations, because each relation corresponds to a one-to-one mapping between the head and tail entities. In other words, TransE assumes that each relation connects one specific head entity to one specific tail entity. To address this issue, TransH [24] selects a mapping plane where each relation (r) to a vector on the plane, adding no extra computational complexity while being effective in handling many-to-one or many-to-many relations. TransH projects entities onto relation-specific hyperplanes before computing the distance between them.

Both TransE and TransH have a common flaw: they embed entities and relations in a space of same dimensions, disregarding the fact that types of entities and relations are different.

TransR [15] was proposed to tackle this limitation by mapping entities and relations to distinct spaces, using a mapping matrix to map the entity space onto the relation space.

Previous literature has only focused on a single aspect of the graph, leading to limited learning capabilities. Enhancing relations by adding more features will increase their complexity and make them harder to learn. Although deep learning methods have a large number of parameters, they have the capability to learn complex relations. The first networks used in the field of graph embedding were Convolutional Neural Networks (CNNs). ConvE [19] and ConvKB [20] are both knowledge graph embedding models that use 2D convolutional neural networks (CNNs) to learn representations of entities and relations in a knowledge graph. While ConvE [19] focuses on learning embeddings that preserve structural and semantic information, ConvKB [20] is optimized for link prediction tasks in knowledge graphs.

Graph Convolution Networks (GCNs) [22] were designed specifically to process graph structures and edges represent relations or connections between entities [25]. The R-GCN is an example of

a GCN that uses an encoder and decoder structure to estimate missing features and score triplets, respectively [26]. The other Convolution Network methods like fast-GCN [22] is an accelerated version of Graph Convolutional Networks (GCNs), a type of neural network architecture designed for learning from graph-structured data. Fast-GCN is specifically optimized for training large-scale graph neural networks efficiently. Fast-GCN improves the scalability and training speed of traditional GCNs by introducing techniques such as neighborhood sampling and parallelization. TransGCN [27] is a specialized variant of Graph Convolutional Networks tailored on graph-structured data, effectively leveraging both node features and graph topology to make predictions for unlabeled nodes in the graph. In SACN [28], self-attention mechanisms are used to capture long-range dependencies and relationships between elements in the input data, such as words in a sentence or nodes in a graph. However, this network does not consider the concepts of relations of two nodes, and gradient loss is a common issue that can cause traps in local optima because the weights of deep learning models are updated by changes in the gradient, if the gradient doesn't change, the weights won't update, and the network won't learn new weights, this is cause of overfit. Additionally, while convolution-based structures have the advantage of learning global features, they are not suitable for non-mesh graph structures, leading to subpar results in methods relying on CNNs. Graph neural networks (GNNs) are based on recurrent neural networks (RNNs). They are deep learning models that utilize neighboring nodes and d-dimensional vectors. In RAGAT [29], an attention-based layer is incorporated, focusing on relation-based estimation functions. RAGAT [29] assigns significant importance to relations. Another example is ID-GNN [30], which classifies the link prediction problem and addresses it as a conditional classifier. ID-GNN [30] underscores the learning of invariant representations. However, in this method, nodes and edges are represented in the same vector space, neglecting the distinction between nodes and edges. Like NBFNet [31], it is based on the Bellman-Ford algorithm and learning is performed according to the graph path. This network maps nodes into the Euclidean space. However, since knowledge graphs contain textual data, this process can result in additional errors. Additionally, computing large data incurs high computational costs, which can be a hindrance. Kg-bert [32] is a deep learning model based on the bert base model and can be understood as a bert model-based learner.

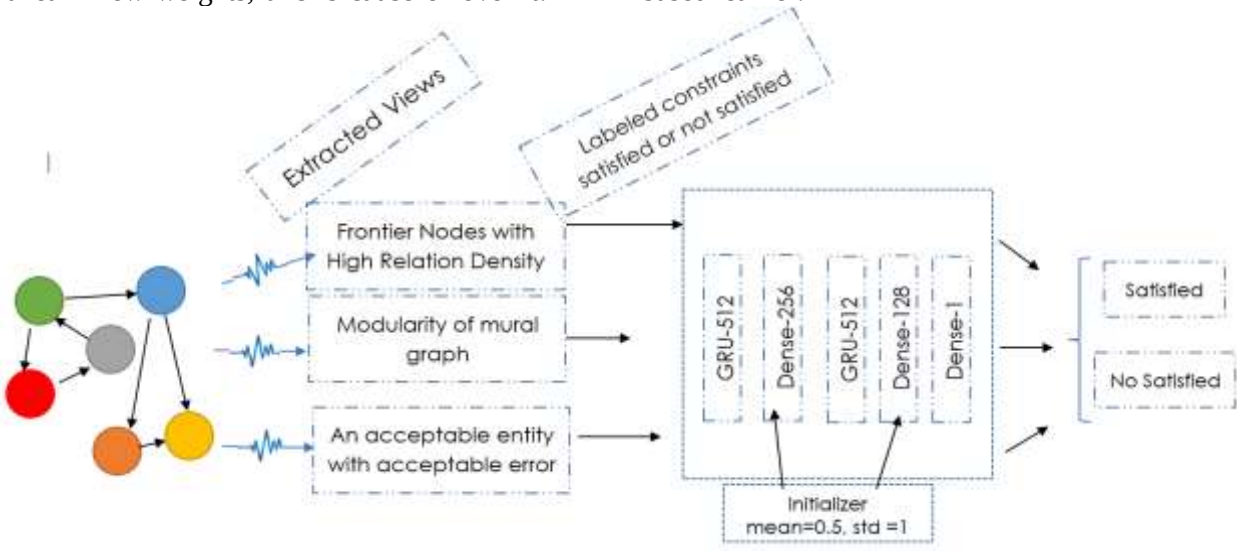

Figure 1. The proposed multi-view CSP, the proposed views are extracted from graph and each feature vector is labeled to satisfy or not satisfied. If feature vector is defined in view so it is satisfy for that view. The GRU-RNN units extract local features and dense units extract global features.

3. Proposed method

The discriminant analysis can used in dependence structures and the dependencies can approximated by graph structures [33], so the structures of graph are used for the link prediction in this paper. The knowledge graph has a complex structure due to the presence of numerous entities and relationships, making the prediction function complex and with a large number of parameters.

In the proposed method, multi-view learning is utilized, where each view has its own prediction function, making it a simple function. In the proposed method, each prediction function

computes a value that shows cost of a triplet (head, relation, tail) or (h, r, t) in each view.
Additionally, the proposed prediction functions can approach the problem from multiple view, which can either be complementary or conflicting.
We proposed the prediction functions learned by a deep learning model.
The complex deep learning models have a high number of parameters and it is the cause of gradient loss so, as a result, an effort of proposed method is made to simplify the network by using a smaller network with reduced parameters. If a deep model has a low number of parameters, it utilizes simpler functions, thereby reducing the occurrence of gradient loss.
The final step is combining the learned functions from separate views to develop a robust estimation function. To achieve this, constraint satisfaction methods were utilized, where each learned function is treated as a constraint to be satisfied. This method is depicted in Figure 1.
The prediction function is learned by the GRU-RNN model and these learned functions get a cost value for test values. If the cost value is bigger than 0.5 so the triplet is accepted. The learned cost function is a constraint that must be satisfied.

4. Constraint Satisfaction Problem(CSP)

The relations in a knowledge graph can be 1-N, N-N, or 1-1. Therefore, a triplet (h, r, t) with a missing value like (h, r, ?), (?, r, t), or (h, ?, t) can have multiple acceptable values for the missing element. Consequently, the proposed method must be capable of finding accepted values.
A CSP is defined as a triplet (V, D, C), in which $v = \{v_1, \dots, v_n\}$ is the set of variables, and $D = \{domain(v_1) \dots domain(v_n)\}$ is the acceptable definition domain for each variable v. Moreover, $C = \{c_1, \dots, c_k\}$ is the set of constraints [34]. Each problem has a set of predefined constraints. Each constraint involves some subset of the variables and specifies the allowable combinations of values for that subset. A state of the problem is assigning a value S_{v1} to a variablev_1, $A = \{v_1 = S_{v1} \dots v_n = S_{vn}\}$ to meet the constraints$C$. The set A is a solution for the CSP. The solution of a CSP can be get from a mathematic formula and solutions have different types, that we introduce two of them.
In a CSP, the goal is to find a solution that satisfies a set of constraints, subject to certain variables taking on specific values. SAT, short for Boolean Satisfiability Problem, is a specific type of Constraint Satisfaction Problem (CSP). In the case of SAT, the variables are Boolean variables, meaning they can take on either true or false values [34,35]. ASP is particularly well-suited for problems that involve finding solutions that satisfy a set of constraints. In ASP, a problem is modeled as a logic program consisting of rules and facts. The goal is to find the answer sets, which represent possible solution to the problem.
The ASP model is employed because graph embedding relies on the assumption that there are N true answers, which need to be analyzed to determine if they meet the constrained conditions. Additionally, each view in the proposed multi-view approach represents a rule or fact that must be satisfied. Each fact, rule, or view is assigned a probability, indicating the ability of a triplet (h, r, t) to satisfy constraints in a solution. Consequently, the constraint estimation function considers the variables of heads, tails and relations.

5. Multi-view Learning

5.1. Multi-view CSP

In the case of a triplet with a missing value, such as (H, R, ?), (?, R, T), or (H, ?, T), the proposed method aims to determine the correctness of a complete triplet by satisfying constraints. The proposed Multi-view CSP is defined as a triplet (V, D, C), where V represents a proposed value from the set of entities and relations. This set is computed using Word2Vec and is called Domain or D. In proposed method, the score functions are considered as constraints or C so, a correct triplet has the highest value. If F_1, F_2, and F_k are constraints, views or rules defined as equation 1, then:

$$\begin{aligned} F_1 &= f_1(head_i, relation, tail) \\ F_2 &= f_2(head_i, relation, tail) \qquad C = \{f_1, f_2, \dots, f_k\} \\ F_k &= f_k(head_i, relation, tail) \qquad P_c = p_1 + p_2 + \dots + p_k \end{aligned} \tag{1}$$

The proposed method evaluates the acceptance of a triplet (h, r, t) from (V, D, C), where C represents constraints, based on whether it satisfies the intended constraints. P_i represents the probability of satisfying a constraint i and P_c is the probability of satisfying all constraints.
The numeric similarity function is not suitable for natural language processing (NLP) tasks because words often have semantic relationships, but their numeric representations may be distant from each other. The structure of graph has semantic relation [33]. So in proposed method used from structure of graph.

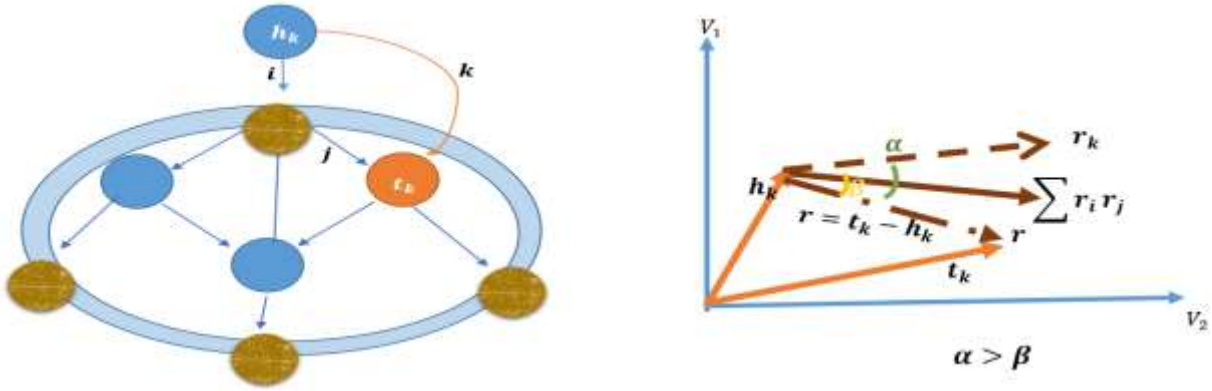

Figure 2. The dominance frontier set of the root node is illustrated. The frontier nodes are on blue ring. It shows that every other node must be connected to at least one node in the dominance frontier node set. If r is the truth value and r_{ij} is most similar to r and r_{ij} is in frontier set. We never get r, we can just get close to r.

Three constraint as views are considered:

- If two nodes are far apart in the graph structure with multiple edges, their straight connection may weaken, thus alternative criteria like stride or proximity to a shared relation should be taken into account.
- A new node can be related to a set of nodes if all nodes in the set and the new node share a similar concept. The similarity of concepts is computed using a newly proposed conceptual clustering method.
- Each relation translates a head to a tail. This translation can have errors that must be considered.

5.1.1. VIEWI: Frontier Nodes with High Relation Density

A path is acceptable if the path between two nodes isn't so long. The presence of a path between two nodes in a graph signifies a relationship, but navigating deep into a knowledge graph can cause disorientation [36]. The long distance between two nodes can result faults in proposed new node because, the meaning or relevance of intermediate nodes becomes less clear or meaningful. This can result in inaccuracies or inconsistencies in knowledge representation and inference. So, in the proposed method, the path between a node and the frontier nodes is used instead of directed path. The dominated subset of a node x is a group of nodes that lie on the path of relation between x and other nodes. Finding the optimal dominated set for every node in a graph is a computationally challenging problem because there exists a large number of possible subsets of vertices, and it is impossible to search through all of them. The dominance frontier is a set of nodes in which node x dominates the parents of the dominated set, meaning x must pass through these nodes to reach other nodes [37]. Therefore, the nodes present on the dominance frontier are used instead of computing the distance between source node and target node. The first step involves determining the dominated set and dominance frontier nodes for each node, as shown in left side Figure 2. Then, the distance from each node to the set of its own frontier nodes is calculated. Equation 2 is utilized to determine the node similarity of path from source node to target node. Let $P = (v_1, v_2, \dots, v_n)$ be a path from v_1 to frontier node v_n in the graph, where $v_1 = (h_1, r_1, t_1)$ and t_1 is considered h_2 forv_2. The path weight is:

$$D_p = \sum_{i=1}^{n-1} \cos Sim(v_i, v_{i+1}) \quad (2)$$

Where p is a random path from h_1 to frontier node t_n and n is frontier node. In the proposed method, we consider r as the best value for distance h and t because embedding create a minimum error for text data. Additionally, r_{ij} is a correctly defined relation, but it can have a minimum distance from r due to the error. r_{ij} is the most similar to r than r_k because frontier nodes are in path of connection of each node. The dominance frontier nodes are identified as critical communication points. They are crucial in establishing connections with other nodes. In order to add a tail to a head, the head must be linked to at least one of these frontier nodes. Thus, the distance to the frontier nodes is calculated instead of finding the distance to a head node.

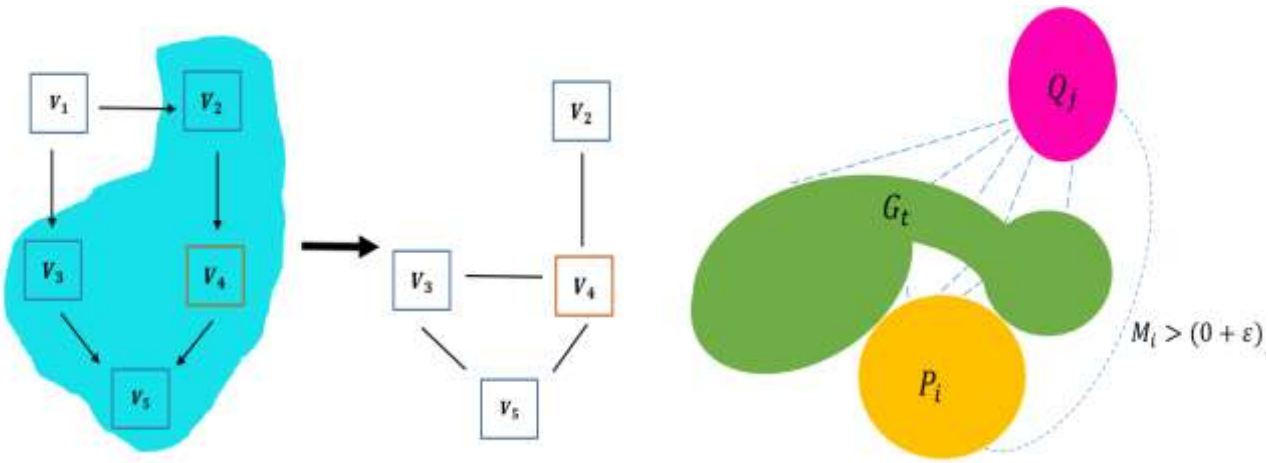

Figure 3. The left side blue area shows Markov blanket of node v4 and moral graph of it is computed, v4 and connected nodes have powerful connection. The modularity (M) is shown so that each of module has powerful internal connection. M has high value for colorful modules.

5.1.2. IEWII: Module Development with Highest Probabilities

Each module has powerful internal connection and modularity is value of dependency of each node to its own module. Conditional independence relationships are indeed powerful in probabilistic modeling. They provide a compact and expressive way to represent complex dependencies among random variables. So, in the proposed method conditional independence relationships are considered for modularity.

A moral graph is used to capture conditional independence relationships implied by the original directed graph. Moral graph forms an edge between every pair of unconnected nodes that share a common child [39]. Each node in the moral graph is connected to its Markov blanket, which the Markov blanket of a node is a set of nodes that fully shields the node from the influence of all other nodes in the network. Nodes outside the Markov blanket are conditionally independent of the node given its Markov blanket. This property is crucial for probabilistic inference and reasoning in Bayesian networks, as it allows for efficient computation of conditional probabilities and predictions.

The moral graph is a way to compute the Bayesian network because moral graph computes conditional independence relationships [38, 39]. In summary, the moral graph serves as a valuable intermediate representation for computing the structure of a Bayesian network from observational data or expert knowledge.

In the proposed method, the conditional independence relationships graph is computed. Then the modules of graph are computed. The modularity of the graph can be calculated through equation 3. A module in a graph or network is a segment with strong relationships, and modularity is a metric for evaluating the quality of a module as shown in figure 3. The weights of each edge are computed by equation 3 and w_{Module} is the modularity number.

$$W_{ij} = \left(\frac{1}{\sqrt{d_i d_j}} + \frac{d_i d_j}{L+1}\right)\delta(C_i C_j)$$
$$W_{module} = \sum_{ij} w_{ij} \quad (3)$$

Where d_i is the degree of node i, L is a constant, δ is a similarity function, $\frac{d_i d_j}{L+1}$ is probability of a common edge between node i and node j, and $\frac{1}{\sqrt{d_i d_j}}$ is the normalized Laplacian graph. According to Figure 3, nodes are divided into classes with the nearly equal cumulative probability after modulation. The nodes of each module have higher probability of mutual relations because the calculated probability is based on the similarity of nodes. In figure 3, the blue lines indicate weak relations of nodes, for they have the minimum probability. The relations of nodes both between and inside modules are based on the probability of similarity and the probability of node relations, something which is very important.

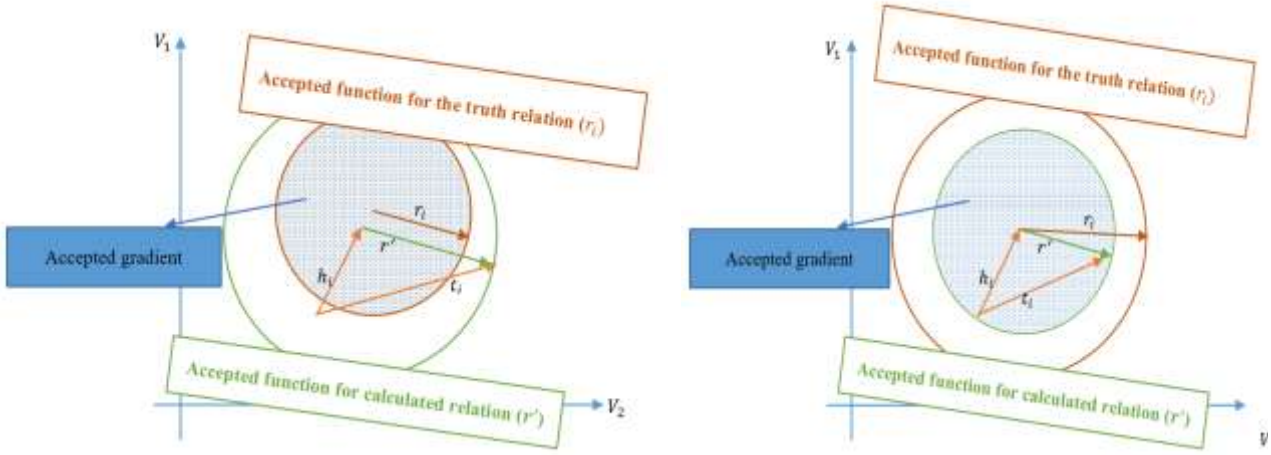

Figure 4. the accepted gradient area.

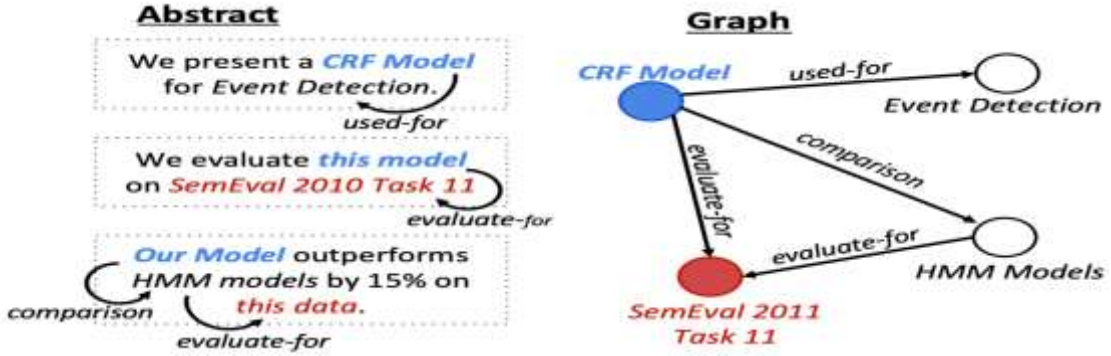

Figure 5. how to change a text to a graph[36].

5.1.3. VIEWIII: Acceptable Range of Triplets

The text analysis error is partly due to mapping a text onto a numerical space, as this transform is definitely prone to a fault. This fault occurs because the embedding of text is not related to the concept of text. Hence, the designed view indicates a range in which each of the interconnected triplets can be true with an error rate. According to Figure 4, the ŕ in green and r in red indicates an embedded relation for head and tail, r_i denotes the accurate and correct embedded relation without fault whereas, ŕ is computed value. Since mapping to a numerical space entails an error rate, ŕ and r_i are distant. So if distant was irreducible, then the answers in this distance are acceptable. The gradient of a value means acceptable changes. Therefore, a reliable range for defining a gradient is indicated by the blue dotted line where the resultant estimation functioning between a head and a tail has the lowest error rate. This range is equal to $\min(r_i, \acute{r})$.A relation is correct and acceptable if it indicates the relation between a head and a tail properly.

If $A = \pi(r^2)$ and if r is radius and $r' = |t_i - h_i|$. $r_i = \alpha r'$. So $A = |t_i - h_i|^2$ or $A = \alpha(|t_i - h_i|)^2$. Then the acceptable gradient area is $|t_i - h_i| \cap \alpha(|t_i - h_i|)^2$ that, $0 < \alpha < 1$ or $\alpha > 1$.

5.2. GRU Network: Learner function

The interconnected and relationship of nodes in a knowledge graph are crucial because a group of nodes along a path can serve as a complete representation of a certain concept (as shown in Figure 5). Thus, a graph with all connected nodes represents a text with a distinct idea. Neglecting the interdependence of nodes can result in incorrect outcomes, but taking into account the relevant sequence of nodes can lead to improved results. As a result, the proposed method for learning should take into account the sequence of data.

The proposed learning model uses a Recurrent Neural Network (RNN), which has the advantage of a memory that enables it to remember its previous learnings. To improve its performance, two advanced versions of RNN, RNN-LSTM [41] and RNN-GRU [42] have been developed. RNN-

LSTM is mainly used for processing texts and videos, but it can suffer from overfitting and gradient loss due to the high number of parameters. It is suitable for long continuous data. To address these issues, RNN-GRU was developed with similar capabilities as RNN-LSTM, but with fewer problems. Therefore, the proposed method uses the GRU-RNN for feature extraction and learning. The GRU consists of a reset gate and an update gate responsible for updating and changing the data. The GRU-RNN is used in all three views of the proposed method. Each GRU layer extracts features, while fully connected layers are responsible for learning the features. The input label in this model is the incidence probability of a triplet, which ranges between 0 and 1. The bias is set to a normal distribution with a mean of 0.5 and a standard deviation of 1. This ensures that all inputs have a proportional probability at the start. Table 1 reports the fine-tuned network parameters. All of the views calculate a score function for the correctness of each triplet. Therefore, the learning problem can be formulated as a regression problem, which can be solved by GRU networks.

Table 1. the GRU-RNN train parameters.

Kernel initializer	Random normal
dropout	0.2
bias	RandomNormal(mean=.5, stddev=1)
activation	Linear, tanh
learning_rate	0.00001
epochs	500
batch_size	50-70
loss	mean_absolute_error

6. Experiments

We describe experimental settings and report empirical results in this section.

We evaluate our model in the KG link prediction task. All experiments are performed on Intel Core i7-7700K CPU@4.20 GHz and NVIDIAGeForce GTX 1070 GPU, and implemented in Python using Anaconda.

6.1.Experiment Setup

Link prediction means proposing an appropriate node in a triplet (?, relation, tail) or (head, relation, ?) where the graph is missing a node. A true triplet indicates a correct concept. Relation prediction means predicting the missing relation in a triplet (head, ? , tail)

Triple Classification Triple Classification is a binary classification task that determines whether a triplet is true or not. In the proposed method, a triplet is evaluated by a multi-view CSP. We measure the number of correct predictions.

In the training phase, each view assigns a score to each triplet and the score functions and their accuracy are learned by GRU networks and a multi-view CSP. Each score function in a view is considered as a constraint that must be satisfied.

In the test phase, random candidates are generated for the missing entity in a triplet (head, relation, tail) in each iteration. Then the deep learning network assigns a score to each random triplet. The score indicates the likelihood of a triplet being true. The true candidate will have the highest score

FB15k-237, WN18RR, and WN18 are the three datasets used in this experiment.

Table 2 shows the number of entities and relations. The Freebase dataset has more relations than entities, but WordNet has more entities than relations. The link prediction is evaluated in two settings: raw and filtered. In the raw setting, the rank of the correct candidate is computed without removing any other correct candidates. In the filtered setting, the ranks are computed after removing the other correct candidates. This paper used the filtered setting.

6.2. Evaluation parameters

The proposed method was evaluated using two groups of parameters: classification accuracy parameters and suggested answer ranking parameters.

- The hit rate is calculated as the total number of attempts made to obtain a suitable answer from a proposed list of length k. The variable $rank_t$ represents the rank of the answer among T proposed answer. The indicator function $I[rank_t < K]$takes a value of 1 if the rank of the suggested answer is less than k.

$$hit@k = \frac{1}{|T|}\sum_{t<|T|} I\left[rank_t < K\right] \quad (4)$$

- Mean Reciprocal Rank (MRR), also known as Average Reciprocal Hit Ratio (ARHR), represents the average approximate location of the best answer. The variable 'rank' indicates the rank of the correct answer.

$$MR = \frac{1}{|T|}\sum rank \quad (5)$$

Table 2. Statistics of data set.

	Entity	relation	Triples		
			train	Valid	test
WN18RR	40943	11	۸۶۰۰۰	3034	3134
WN18	40943	18	۱۴۰۰۰۰	5000	5000
FB15k-237	14541	237	200000	17535	20466

Table 3. Experimental result on test date set.

Method	WN18		FB15k-237		WN18RR	
	@hit10	MR	@hit10	MR	@hit10	MR
DistMult	94.61	675	41.9	199	50.22	1107
ComplEX	96.15	190	42.8	144	71.29	793
TransE	94.87	279	42.0	209	67.39	1187
RotateE	96.02	309	53.3	178	57.35	3318
ConvE	95.68	504	50.1	281	47.62	281
R-GCN	95.5	-	41.7	-	-	-
KBGAN	94.9	-	45.8	-	48.1	-
KG-BERT	-	-	42.00	153	52.4	97
Proposed method	97.04	56.6	94.13	50.7	88.81	66.1

In the test phase, we compute the CSP multi-view score for the true and random candidates. Then we compute the quality metrics. We repeat this process several times and report the average values.

Table 3 shows the Link Prediction performance of the state-of-the-art models. We test the proposed model using the pykg2vec toolkit. The proposed model achieved a high Hits@10 score and a low MR score on the test datasets. Hits@10 measures the proportion of correct entities in the top 10 candidates. A better method has a low MR and a high Hits@N. The proposed method models the structure of the graph and the semantic relations. We use the proposed geometrical learners as views, and the views are learned by the RNN-GRU deep neural network. The views are integrated by the CSP multi-view.

The values in Table 3 are the averages. The MR value is the lowest possible rank of the true answer and the Hit rate is the acceptable range. As shown in Table 3, the proposed method has the lowest values in hit rate and MR.

The triple classification means whether a triplet is correct or not. We compute the correctness of a random head and a random tail with a given relation.

That means, given a true relation, a random head and tail are generated. This method is used for triple correctness calculation. The comparison of the proposed method and the state-of-the-art models is shown in Table 4.

The number of parameters used is shown in Table 5. The values in Table 3, Table 4, Table 5, are from [1, 32, 43, 9]. The higher hit and lower MR are better. The comparison between the proposed method and the state-of-the-art methods is shown in Table 3.

As shown, the proposed method has the best performance on the evaluation metrics. WordNet dataset has fewer relations than entities, so the triples are not unique and link prediction in knowledge graph embedding is very hard. The structure of the graph can be captured by the individual triplets. It seems that the more distinct the relation is, the more likely the learning is.

Table 4. Evaluation results of triple classification.

Method	WN18RR	FB15k-237
TransE	75.9	79.8
TransR	85.9	82.1
DistMult	87.1	86.2
ConvKB	87.6	88.8
Kg-Bert	93.5	90.4
Proposed method	97.0	91.36

Table 5. Total parameter.

method	Total parameter(million parameter)
DKRL(CNN)	9.3
LSTM	11.2
GCN	226
Proposed method	2.3

For this reason, the proposed method has the lowest performance on WN18RR than the other datasets. Table 4 shows the evaluation of triple classification. The proposed method has the best performance. In this method, a relation is given and a suitable head and tail are determined. The entities (head and tail) are generated randomly and a score is computed for each pair. A triplet with a high score will be correct.

Table 3, Table 4, and Table 5 show that the proposed method has no inconsistent results compared to the other models. The number of parameters used is shown in Table 5. The proposed method uses several simple geometric learners as views. Therefore, the proposed model (GRU-RNN) has a simple architecture and a low number of parameters. The training parameters are shown in Table 1.

The proposed model has the best performance on the evaluation metrics and it has features such as low number of parameters for simple structure, low complexity for simple architecture, consistency among evaluation metrics, expandability because the proposed method has separate learner functions, and simple architecture.

7. Conclusion

In this study, we present a multi-view CSP approach based on deep learning for link prediction in knowledge graph embedding. Our results demonstrate that by combining geometric features with semantic features, we can improve the accuracy of the score function estimation. Each view in our approach captures a unique geometric feature of the graph. The first view measures the similarity of a node to the boundary nodes, which are the communication bottlenecks. The second view focuses on semantic similarity, using a Laplacian graph to represent the connections in the graph. The final view takes into account the error rate of the embedding score function. We use the GRU-RNN model to learn the score function for each view, and we treat each view as a constraint that must be satisfied. One of the limitations of existing models is the low accuracy when the number of relations and entities is imbalanced.

Our experiments show that our proposed model outperforms these models on the evaluation metrics. Future research will mainly address the optimization of manifold learning by concept learning. The knowledge graph has meaningful short sentences that must reflect the true concepts.

Reference

[1] A. Rossi, D. Barbosa, D. Firmani, A. Matinata.: Knowledge graph embedding for link prediction: A comparative analysis. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 2021, vol. 15, pp. 1-49.

[2] A. Saxena, A. Tripathi, P. Talukdar.: Improving multi-hop question answering over knowledge graphs using knowledge base embeddings. *In Proceedings of the 58th annual meeting of the association for computational linguistics*, 2020.

[3] H. Liu, H. Kou, C. Yan, L. Qi.: Link prediction in paper citation network to construct paper correlation graph. In *Wireless Communications and Networking*, 2019, vol. 1, pp.1–12.

[4] L. Labiod, M. Nadif.: Efficient regularized spectral data embedding. *Advances in Data Analysis and Classification*, 2021, vol. 15, pp.99-119.

[5] B. Jagvaral, W.K. Lee, J.S. Roh, M.S. Kim, et al.: Path-based reasoning approach for knowledge graph completion using CNN-BiLSTM with attention mechanism. *Expert Systems with Applications*, 2020, vol. 142.

[6] Q. Zhang, R. Wang, J. Yang, L. Xue.: Knowledge graph embedding by translating in time domain space for link prediction. *Knowledge-Based Systems*, 2021, vol. 212.

[6] X. Chen, S. Jia, Y. Xiang.: A review: Knowledge reasoning over knowledge graph. *Expert systems with applications*, 2020, vol. 141.

[7] S. Ji, S. Pan, E. Cambria, P. Marttinen, et al.: A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE transactions on neural networks and learning systems*, 2021, vol. 33, pp. 494-514.

[8] K. Wang, Y. Liu, X. Xu, Q.Z. Sheng.: Enhancing knowledge graph embedding by composite neighbors for link prediction. *Computing*, 2020, vol. 102, pp. 2587-2606.

[9] M. Molaei, D. Mohamadpur.: Distributed Online Pre-Processing Framework for Big Data Sentiment Analytics. *Journal of AI and Data Mining*, 2022, vol.10, pp.197-205.

[10] A. Lakizadeh, E. Moradizadeh.: Text sentiment classification based on separate embedding of aspect and context. *Journal of AI and Data Mining*, 2022, vol. 10, pp.139-149.

[11] A. Popescu, S. Polat-Erdeniz, A. Felfernig, M. Uta, et al: An overview of machine learning techniques in constraint solving. *Journal of Intelligent Information Systems*, 2022, vol.58, pp.91-118.

[12] R. Socher, D. Chen, C.D. Manning, et al.: Reasoning with neural tensor networks for knowledge base completion. *Advances in neural information processing systems*, 2013.

[13] Q. Wang, Z. Mao, B. Wang, L. Guo.: Knowledge graph embedding: A survey of approaches and applications. *IEEE transactions on knowledge and data engineering*, 2017, vol. 29, pp.2724-2743.

[14] Z. Wang, J. Zhang, J. Feng, Z. Chen, et al.: Knowledge graph embedding by translating on hyperplanes. *In Proceedings of the AAAI conference on artificial intelligence*, 2014, Vol. 28.

[15] A. Bordes, N. Usunier, A. Garcia-Duran, et al.: Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 2013.

[16] M. Nickel, L. Rosasco, T. Poggio.: Holographic embeddings of knowledge graphs. *In Proceedings of the AAAI conference on artificial intelligence*, 2016, Vol. 30.

[17] T. Trouillon, J. Welbl, S. Riedel.: Complex embeddings for simple link prediction. *In International conference on machine learning*, 2016, pp. 2071-2080.

[18] T. Dettmers, P. Minervini, P. Stenetorp.: Convolutional 2d knowledge graph embeddings. *In*

Proceedings of the AAAI conference on artificial intelligence, 2018, Vol. 32.

[19] D.Q. Nguyen.: A novel embedding model for knowledge base completion based on convolutional neural network. *arXiv preprint arXiv:1712.02121*, 2017.

[20] J. Chen, X. Feng, L. Jiang, Q. Zhu.: State of charge estimation of lithium-ion battery using denoising autoencoder and gated recurrent unit recurrent neural network. *Energy*, 2021, vol. 227.

[21] J. Chen, T. Ma, C. Xiao.: Fastgcn: fast learning with graph convolutional networks via importance sampling. *arXiv preprint arXiv:1801.10247*, 2018.

[22] L. Cai, B. Yan, G. Mai, K. Janowicz, R. Zhu.: TransGCN: Coupling transformation assumptions with graph convolutional networks for link prediction. *In Proceedings of the 10th international conference on knowledge capture*, 2019, pp. 131-138.

[23] Y. Lin, Z. Liu, M. Sun, Y. Liu, X. Zhu.: Learning entity and relation embeddings for knowledge graph completion. *In Proceedings of the AAAI conference on artificial intelligence*, 2015, Vol. 29, No. 1.

[24] S. Arora.: A survey on graph neural networks for knowledge graph completion. *arXiv preprint arXiv:2007.12374*, 2020.

[25] J. Chen, X. Feng, L. Jiang, Q. Zhu.: State of charge estimation of lithium-ion battery using denoising autoencoder and gated recurrent unit recurrent neural network. *Energy*, 2021, vol. 227, p.120451.

[26] D. Yu, Y. Yang, R. Zhang, Y Wu.: Knowledge embedding based graph convolutional network. *In Proceedings of the web conference* 2021, pp. 1619-1628.

[27] C. Shang, Y. Tang, J. Huang, J. Bi, X. He.: End-to-end structure-aware convolutional networks for knowledge base completion. *In Proceedings of the AAAI conference on artificial intelligence*, 2019, Vol. 33, No. 01, pp. 3060-3067.

[28] X. Liu, H. Tan, Q. Chen, G. Lin.: Ragat: Relation aware graph attention network for knowledge graph completion. *IEEE Access*, 2021, vol.9, pp.20840-20849.

[29] J. You, J.M. Gomes-Selman, R. Ying, et al.: Identity-aware graph neural networks. *In Proceedings of the AAAI conference on artificial intelligence*, 2021, Vol. 35, No. 12, pp. 10737-10745.

[30] Z. Zhu, Z. Zhang, L.P. Xhonneux,: Neural bellman-ford networks: A general graph neural network framework for link prediction. *Advances in Neural Information Processing Systems*, 2021, vol. 34, pp.29476-29490.

[31] L. Yao, C. Mao, Y. Luo.: KG-BERT: BERT for knowledge graph completion. *arXiv preprint arXiv:1909.03193*, 2019.

[32] G. Perez-de-la-Cruz, G. Eslava-Gomez.: Discriminant analysis for discrete variables derived from a tree-structured graphical model. *Advances in Data Analysis and Classification*, 2019, vol. 13, pp.855-876.

[33] F. Jafarinejad,: Benefiting from Structured Resources to Present a Computationally Efficient Word Embedding Method. *Journal of AI and Data Mining*, 2022, vol.10, pp.505-514.

[34] A.L. Tesfaye.: Constrained Dominant sets and Its applications in computer vision. *arXiv preprint arXiv:2002.06028*, 2020.

[35] J. Hartog, H.V. Zanten.: Nonparametric Bayesian label prediction on a graph. *Computational Statistics & Data Analysis*, 2018, vol. 120, pp.111-131.

[36] S.W. Cunningham, J.H. Kwakkel,: Technological frontiers and embeddings: A visualization approach. *In Proceedings of PICMET'14 Conference: Portland International Center for Management of Engineering and Technology; Infrastructure and Service Integration*, 2014, pp. 2891-2902.

[37] D.F. Jerding, J.T. Stasko.: The information mural: A technique for displaying and navigating large information spaces. *IEEE Transactions on Visualization and Computer Graphics*, 1998, vol. 4, pp.257-271.

[38] Y. Li, K. Korb, L. Allison.: The complexity of morality: Checking markov blanket consistency with DAGs via morality. *arXiv preprint arXiv:1903.01707*, 2019.

[39] https://lyusungwon.github.io/studies/2019/09/19/graphwriter.

[40] A. Sherstinsky.: Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Physica D: Nonlinear Phenomen*a, 2020, vol. 404.

[41] R. Dey, F.M. Salem: Gate-variants of gated recurrent unit (GRU) neural networks. *In IEEE 60th international midwest symposium on circuits and systems*, 2017, pp. 1597-1600.

[42] M. Schlichtkrull, T.N. Kipf, P. Bloem, et al: Modeling relational data with graph convolutional networkss. *In European semantic web conference*, 2018.

[43] L. Cai, W.Y. Wang.: Kbgan: Adversarial learning for knowledge graph embeddings. *arXiv preprint arXiv:1711.04071*, 2017.

یک مدل چندنمایی برای پیش‌بینی ارتباط در تعبیه‌سازی گراف دانش با استفاده از GRU-RNN به عنوان مساله ارضای محدودیت

افروز مرادبیگی و فرزین یغمایی*

دانشکده برق و کامپیوتر، دانشگاه سمنان، سمنان، ایران .

ارسال ۲۰۲۳/۱۱/۲۵؛ بازنگری ۲۰۲۴/۰۲/۲۷؛ پذیرش ۲۰۲۴/۰۴/۰۸

چکیده:

گراف دانش ابزار پرکاربردی در زمینه استدلال است، بطوریکه استدلال از طریق پیش‌بینی پیوند در نمودار دانش تسهیل می‌شود. با این حال، روش‌های سنتی دارای محدودیت‌هایی مانند پیچیدگی بالا یا ناتوانی در ثبت مؤثر ویژگی‌های ساختاری گراف هستند. چالش اصلی در مدیریت همزمان هر دو ویژگی ساختاری و شباهت گراف نهفته است. در این مطالعه، ما از یک رویکرد ارضای محدودیت استفاده می‌کنیم، که در آن هر پیوند پیشنهادی باید هم محدودیت‌های ساختاری و هم محدودیت‌های شباهت را برآورده کند. برای این منظور، هر محدودیت از منظر خاصی در نظر گرفته می شود که از آن به عنوان نما یاد می شود. هر نما یک امتیاز احتمال را با استفاده از یک GRU-RNN محاسبه می‌کند، که محدودیت از پیش تعریف شده خود را برآورده می‌کند. در محدودیت اول، گره پیشنهادی باید احتمال بالای ۰٫۵ با گره‌های مرزی داشته باشد. محدودیت دوم گراف بیزین را محاسبه می‌کند و گره پیشنهادی باید پیوندی در گراف بیزین داشته باشد. آخرین محدودیت مستلزم آن است که یک گره پیشنهادی می‌تواند یک خطای قابل قبول قرار داشته باشد. این اجازه می‌دهد تا روابط N-N به طور دقیق تعیین شود، در حالی که محدودیت‌های تعبیه سازی را نیز مورد توجه قرار می‌دهد. نتایج آزمایش‌ها نشان داد که روش پیشنهادی عملکرد دو مجموعه داده استاندارد را بهبود می‌بخشد.

کلمات کلیدی: چندجانبه، تعبیه سازی گراف دانش، مساله ارضای محدودیت، GRU-RNN.